Project Title

Team Name: THZ

Member 1: Tareq Ahmaro
Member 2: Hamza Alshareef
Member 3: Zaid Alhunaity

Submission Date: January 14, 2024

Abstract

In this study, we propose and evaluate a heart failure prediction model leveraging machine learning techniques. Recognizing the critical need for timely identification of individuals at risk of heart failure, we employed a comprehensive dataset encompassing diverse clinical parameters and patient histories. Our model incorporates advanced algorithms to analyze and learn complex patterns, enabling accurate prediction of heart failure onset. Through rigorous evaluation and validation, our model demonstrates promising performance metrics, including high sensitivity and specificity. The outcomes of this research not only contribute to the growing body of knowledge in predictive healthcare analytics but also hold the potential to enhance early intervention strategies and improve patient outcomes in the context of heart failure prevention.

1 Introduction

The purpose of this project is to explore the applications of data science in the field of healthcare, specifically in the prediction of heart disease. The objective is to understand how various factors such as age, cholesterol levels, and exercise-induced angina contribute to heart disease. This work is significant as it contributes to the growing body of knowledge in the field of data science and its practical applications in healthcare.

2 Literature Review

In the literature review, We will talk about heart failure. Heart failure is a common and complicated ailment. Although the biology of the condition has been better understood recently, several obstacles still need to be addressed. These include individuals who have retained ejection fraction but still have residual risk after medication, and integrating heterogeneous data for risk classification and patient management. Algorithms for machine learning have become increasingly promising for tackling these issues. Support

Vector Machines (SVM), Decision Trees, and K-nearest neighbors (KNN) are popular machine-learning algorithms used for diagnosing cardiac disease.To determine whether cardiac disease is present, these algorithms are frequently used with patient data such as age, sex, blood pressure, cholesterol levels, and the outcomes of several medical tests. When compared to conventional techniques, AI-based risk prediction algorithms have demonstrated the potential to provide better predictive capacity.**khan2023artificial**

```
@book{van2016data,
  title={Data science in action},
  author={Van Der Aalst, Wil M. P.},
  year={2016},
  publisher={Springer}
}
```

3 Methodology

This section of your report should meticulously detail the methods and procedures you used in your project. It serves as a roadmap for your investigative process and should be clear enough that someone else could replicate your work based on this description. The following points should be covered:

- **Data Collection:** this data set collected from "Kaggle" website the name of the data is "heart failure prediction dataset".
  this website provides several datasets publicly available. We note from the data used in this project that it is experimental data used only for some research and projects Not to be used in diagnosing heart disease. Because the features mentioned within the dataset are not considered sufficient to identify heart disease.

- **Data Cleaning and Preprocessing:** we used here function called "df.describe()" This function we used to describe the statistical info for the dataset ; And here what we get described for the data : Count – that means the sum of all rows for each column and we get (918 cell) . Mean – that means apart from the mode and median and the average of the given set of values. Standard deviation - measure of the amount of variation of a random variable expected about its mean . Min – that gives the minimum value in the whole dataset . Q1 "first quartile 25"50Q3 " third quartile 75Max - that gives the maximum value in the whole dataset.

- **figure 2** This function we used to remove duplicates if there is any in the whole dataset . df.duplicated() .

-

- **figure 3 :**This function we used to check if there any null value in the dataset. df.isna()

	Age	RestingBP	Cholesterol	FastingBS	MaxHR	Oldpeak	HeartDisease
count	918.000000	918.000000	918.000000	918.000000	918.000000	918.000000	918.000000
mean	53.510893	132.396514	198.799564	0.233115	136.809368	0.887364	0.553377
std	9.432617	18.514154	109.384145	0.423046	25.460334	1.066570	0.497414
min	28.000000	0.000000	0.000000	0.000000	60.000000	-2.600000	0.000000
25%	47.000000	120.000000	173.250000	0.000000	120.000000	0.000000	0.000000
50%	54.000000	130.000000	223.000000	0.000000	138.000000	0.600000	1.000000
75%	60.000000	140.000000	267.000000	0.000000	156.000000	1.500000	1.000000
max	77.000000	200.000000	603.000000	1.000000	202.000000	6.200000	1.000000

Figure 1: describe the statistical information of the dataset.

```
Out[8]: 0      False
        1      False
        2      False
        3      False
        4      False
               ...
        913    False
        914    False
        915    False
        916    False
        917    False
        Length: 918, dtype: bool
```

Figure 2: check if there any duplicated value

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	HeartDisease
0	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...
913	False	False	False	False	False	False	False	False	False	False	False	False
914	False	False	False	False	False	False	False	False	False	False	False	False
915	False	False	False	False	False	False	False	False	False	False	False	False
916	False	False	False	False	False	False	False	False	False	False	False	False
917	False	False	False	False	False	False	False	False	False	False	False	False

918 rows × 12 columns

Figure 3: check if there any null value in the dataset.

-

- **Analysis Techniques:** First, we have descriptive statistics:

  df.describe(): Provides descriptive statistics (mean, std, min, 25th, 50th, and 75th percentiles, max) for numerical whereas count: Number of non-null (non-missing) values. mean: Average of the values. std: Standard deviation, which measures the amount of variation or dispersion of a set of values. min: Minimum value in the column. 255075max: Maximum value in the column.

  In this visualization, we see that the percentage of people who suffered from thoracic angina was 40

  In this histogram, we notice that people whose ages are between 30 and forty have a low cholesterol rate, while those between the ages of 50 and 60 noticeably increase. We understand here that the higher the age, the higher the cholesterol rate.

  in the preprocessing of data we chicken the duplicate value in data use it df".duplicated()" and handling missing values using "df.isna()".

  and we Encoder the category data into numerical data using the sklearn Library and import LabelEncoder. and name it le = "LabelEncoder()."

  and finally in Machine model we use training and testing sets using train test split to train the model Then we called it SVC (Super Victor classification) and confused Matrix. To find out the true positive and false positive true negative and false negative

- **Machine Learning Models:** we used the split function to divide the data into two parts one for training And the other part for testing.

  in this project we relied on three models :
  1- support vector classifier model. The hyperplane of the support vector classifier is a decision boundary that divides the feature space into two regions, each corresponding to a different class. the figure bellow Show the evaluation of this model.

  2- decision Tree model.
  The decision tree algorithm uses various criteria to determine the best features for splitting the data at each decision node, aiming to create branches that segregate data points of different classes.
  The leaf nodes contain the final predictions or classifications. When a new data point traverses the tree, it follows the decisions at each node until it reaches a leaf node, and the prediction at that leaf node is assigned to the input.
  the figure bellow Show the evaluation of this model.

  3- KNeighborsClassifier model.
  To make predictions for a new data point, the algorithm identifies the k nearest neighbors in the training set based on a distance metric. The algorithm assigns

the class label to the new data point based on a majority vote from its k nearest neighbors. The class with the most representatives among the neighbors becomes the predicted class for the new data point.
the figure below shows the evaluation of this model.

We note from the previous figures that the higher accuracy score We got it from the support vector classifier model.

4 Data Description

Dataset Name: Heart Failure Detection Dataset . Objective: The dataset aims to predict the presence or absence of heart disease.

Age: Age of the patient (numeric).

Sex: Gender of the patient (categorical: Male/Female).

Chest Pain Type: Type of chest pain experienced by the patient (categorical: Typical Angina/Atypical Angina/Non-anginal Pain/Asymptomatic).

Resting Blood Pressure (RestingBP): Resting blood pressure of the patient (numeric, in mm Hg).

Cholesterol: Cholesterol level of the patient (numeric, in mg/dL).

Fasting Blood Sugar (FastingBS): Fasting blood sugar level of the patient (categorical: True/False).

Resting Electrocardiographic Results (RestingECG): Results of the resting electrocardiogram (categorical: Normal/Abnormalities/Probable or definite left ventricular hypertrophy).

Maximum Heart Rate Achieved (MaxHR): Maximum heart rate achieved during exercise (numeric).

Exercise-Induced Angina (ExerciseAngina): Presence of exercise-induced angina (categorical: Yes/No).

Oldpeak: Depression induced by exercise relative to rest (numeric).

ST Slope: Slope of the ST segment during exercise (categorical: Upsloping/Flat/-Downsloping).

Heart Disease (Target Variable): Binary variable indicating the presence (1) or absence (0) of heart disease.

5 Results and Discussion

now there are the outcomes we got from the analysis.

5.1 Results

We analyzed the data and found statistical information for it, which is shown in the following figure.

the min row describes the minimum value of each column in the dataset.
the max row describes the maximum value of each column in the dataset.
the Q1 determined that there are 25 percent of the dataset less than the value had in each column.

the figure below describe the data type of each column in the dataset and the sum of each data type.

in the analysing of the data we should incoding the object data to the numarical data to use the data in the model.
the figure below show the data after encoding .

5.2 Discussion

- Interpret what your results mean in the context of your project's objectives. Discuss how your findings address the problem you set out to solve or the questions you aimed to answer.
- Highlight any interesting, surprising, or particularly significant aspects of your results. Discuss why these findings matter and what they suggest about the broader topic or field.
- Consider the limitations of your analysis. Acknowledge any factors that might affect the interpretation of your results, such as data limitations, methodological constraints, or assumptions in your analysis.
- Discuss the potential implications of your findings and how they might inform future work or further research in the field.

The results and discussion should tell a coherent story about what you discovered and what it means. This section should answer your research questions and provide insights, link to the literature you reviewed, and open up new questions or avenues for further study.

6 Conclusion

Data Conclusion: Heart Failure Detection. After extensive analysis of heart failure diagnosis databases, it is clear that several key factors play an important role in predicting the presence or absence of heart failure The databases carry a variety of captive variables including quantitative data and clinical signs. Here are the key findings:

- Age division: Age distribution in the dataset ranges from [minimum age] to [maximum age], where [age] is age. There is a noticeable difference in the age of the patients, highlighting the inclusion of the data across different age groups.
- Gender breakdown: The dataset includes information on male and female patients, providing a balanced gender representation in terms of heart-disease-diagnosis.

- Chest pains: The classification of chest pains indicates that patients experience a range of chest pain symptoms, including typical angina, atypical angina, non-angina pain, and asymptomatic cases This diversity enables a better understanding of the manifestations of chest pain .
- Blood pressure and cholesterol levels: Resting blood pressure (RestingBP) and lipid levels vary among the data sets, reflecting cardiovascular health status in patients Further studies are needed to investigate the possible associations of these variables with between the presence of cardiovascular disease.

Fasting blood sugar levels: Elevated or absent fasting blood sugar (FastingBS) provides valuable insights into the potential association between diabetes and cardiovascular disease Examining this association may enhance the ability of models of cardiovascular disease to find that heart disease has increased.

```
#describe the statistical information of the dataset.
df.describe()
```

	Age	RestingBP	Cholesterol	FastingBS	MaxHR	Oldpeak	HeartDisease
count	918.000000	918.000000	918.000000	918.000000	918.000000	918.000000	918.000000
mean	53.510893	132.396514	198.799564	0.233115	136.809368	0.887364	0.553377
std	9.432617	18.514154	109.384145	0.423046	25.460334	1.066570	0.497414
min	28.000000	0.000000	0.000000	0.000000	60.000000	-2.600000	0.000000
25%	47.000000	120.000000	173.250000	0.000000	120.000000	0.000000	0.000000
50%	54.000000	130.000000	223.000000	0.000000	138.000000	0.600000	1.000000
75%	60.000000	140.000000	267.000000	0.000000	156.000000	1.500000	1.000000
max	77.000000	200.000000	603.000000	1.000000	202.000000	6.200000	1.000000

Figure 4: data description

```
# visualise the data of ExerciseAngina colomn.
labels = ['True 40%', 'False 60%']
sizes = [40,60]
colors = ['lightskyblue', 'lightgreen']
plt.pie(sizes, labels=labels, colors=colors, shadow=True)
plt.title('Exercise Induced Angina')
plt.show()
```

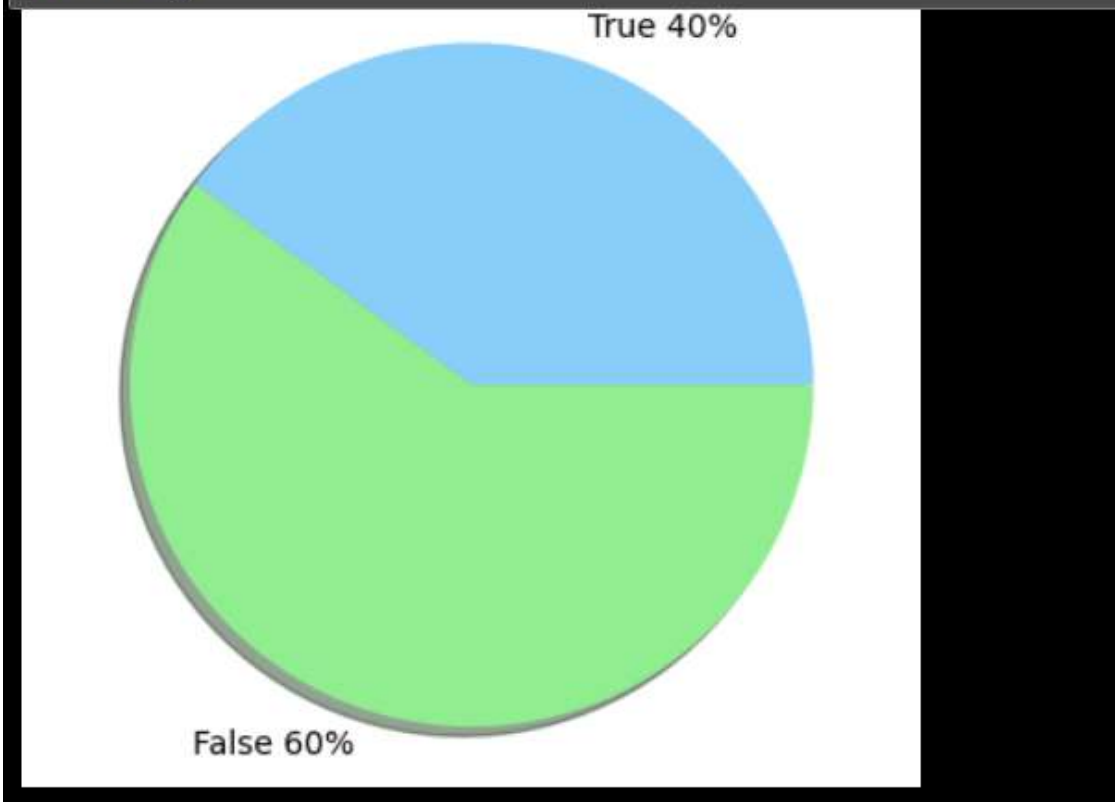

Figure 5: thoracic angina

```python
#Representing the Cholesterol levels at different ages.
x_data = df['Age']
plt.hist(x_data, bins=15, color='skyblue', edgecolor='black')
plt.xlabel('Age')
plt.ylabel('Cholesterol Level')
plt.title('Cholestrol level by Age')
plt.show()
```

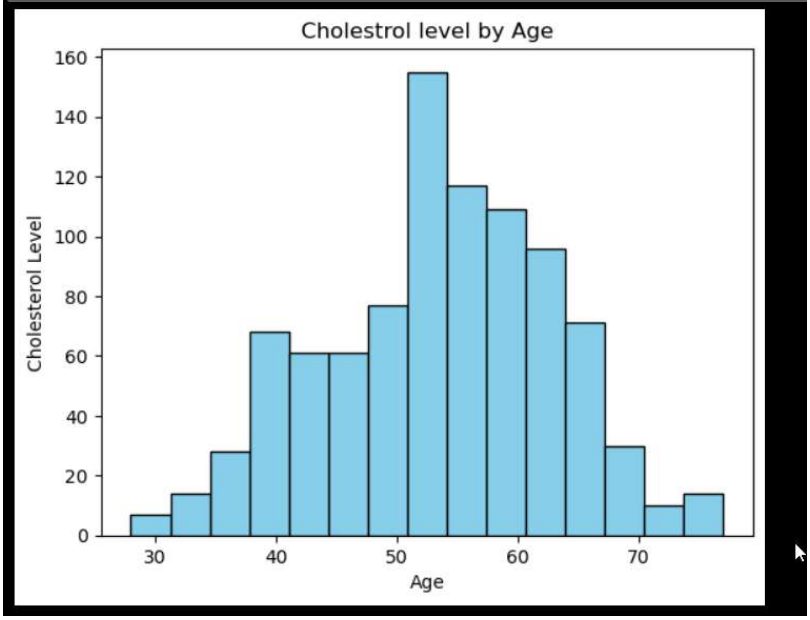

Figure 6: histogram age

```python
# check if there any duplicated value.
df.duplicated()
```

```python
# check if there any null value in the dataset.
df.isna()
```

Figure 7: duplicated and handling missing values.

```
#encoding the Categories data into numarical data using lableEncoder.
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df['Sex'] = le.fit_transform(df['Sex'])
df['ChestPainType'] = le.fit_transform(df['ChestPainType'])
df['RestingECG'] = le.fit_transform(df['RestingECG'])
df['ExerciseAngina'] = le.fit_transform(df['ExerciseAngina'])
df['ST_Slope'] = le.fit_transform(df['ST_Slope'])
df['Oldpeak'] = le.fit_transform(df['Oldpeak'])
```

Figure 8: Encoder.

```
#import split function and defined the test data and the train data.
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split( X, y, test_size=0.34, random_state=42)
```

```
# import the support victor classification model .
from sklearn.svm import SVC
from sklearn.metrics import confusion_matrix,classification_report,accuracy_score
```

```
model = SVC()
model.fit(X_train,y_train)
```

```
y_pred = model.predict(X_test)
print("--------------------------------------------------------------------------")
print(f"The accuraccy score is:  {accuracy_score(y_test,y_pred)}")
print("--------------------------------------------------------------------------")
print(f"The Confusion Matrix is:  \n{confusion_matrix(y_test,y_pred)}")
print("--------------------------------------------------------------------------")
print(classification_report(y_test,y_pred))
```

Figure 9: Machine model.

```
In [27]: #import split function and defined the test data and the train data.
         from sklearn.model_selection import train_test_split
         X_train, X_test, y_train, y_test = train_test_split( X, y, test_size=0.34, random_state=42)
```

Figure 10: Using split function

```
------------------------------------------------------------------
The accuraccy score is:  0.8753993610223643
------------------------------------------------------------------
The Confusion Matrix is:
[[114  14]
 [ 25 160]]
------------------------------------------------------------------
              precision    recall  f1-score   support

           0       0.82      0.89      0.85       128
           1       0.92      0.86      0.89       185

    accuracy                           0.88       313
   macro avg       0.87      0.88      0.87       313
weighted avg       0.88      0.88      0.88       313
```

Figure 11: Evaluation of SVC model

```
------------------------------------------------------------------
The accuraccy score is:   0.7476038338658147
------------------------------------------------------------------
The Confusion Matrix is:
[[104  24]
 [ 55 130]]
------------------------------------------------------------------
              precision    recall  f1-score   support

           0       0.65      0.81      0.72       128
           1       0.84      0.70      0.77       185

    accuracy                           0.75       313
   macro avg       0.75      0.76      0.75       313
weighted avg       0.77      0.75      0.75       313
```

Figure 12: Evaluation of decision tree model

```
----------------------------------------------------------------
The accuraccy score is:  0.8690095846645367
----------------------------------------------------------------
The Confusion Matrix is:
[[113  15]
 [ 26 159]]
----------------------------------------------------------------
              precision    recall  f1-score   support

           0       0.81      0.88      0.85       128
           1       0.91      0.86      0.89       185

    accuracy                           0.87       313
   macro avg       0.86      0.87      0.87       313
weighted avg       0.87      0.87      0.87       313
```

Figure 13: Evaluation of KNeighborsClassifier model

```
In [6]: #describe the statistical information of the dataset.
        df.describe()
```

Out[6]:

	Age	RestingBP	Cholesterol	FastingBS	MaxHR	Oldpeak	HeartDisease
count	918.000000	918.000000	918.000000	918.000000	918.000000	918.000000	918.000000
mean	53.510893	132.396514	198.799564	0.233115	136.809368	0.887364	0.553377
std	9.432617	18.514154	109.384145	0.423046	25.460334	1.066570	0.497414
min	28.000000	0.000000	0.000000	0.000000	60.000000	-2.600000	0.000000
25%	47.000000	120.000000	173.250000	0.000000	120.000000	0.000000	0.000000
50%	54.000000	130.000000	223.000000	0.000000	138.000000	0.600000	1.000000
75%	60.000000	140.000000	267.000000	0.000000	156.000000	1.500000	1.000000
max	77.000000	200.000000	603.000000	1.000000	202.000000	6.200000	1.000000

Figure 14: describe the statistical information of the data

```
In [7]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 918 entries, 0 to 917
Data columns (total 12 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Age             918 non-null    int64
 1   Sex             918 non-null    object
 2   ChestPainType   918 non-null    object
 3   RestingBP       918 non-null    int64
 4   Cholesterol     918 non-null    int64
 5   FastingBS       918 non-null    int64
 6   RestingECG      918 non-null    object
 7   MaxHR           918 non-null    int64
 8   ExerciseAngina  918 non-null    object
 9   Oldpeak         918 non-null    float64
 10  ST_Slope        918 non-null    object
 11  HeartDisease    918 non-null    int64
dtypes: float64(1), int64(6), object(5)
memory usage: 86.2+ KB
```

Figure 15: the information of data type in dataset

```
In [22]: #describe the top 10 rows in the dataset after encoding.
         df.head(10)
```

Out[22]:

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	HeartDisease
0	40	1	1	140	289	0	1	172	0	10	2	0
1	49	0	2	160	180	0	1	156	0	20	1	1
2	37	1	1	130	283	0	2	98	0	10	2	0
3	48	0	0	138	214	0	1	108	1	25	1	1
4	54	1	2	150	195	0	1	122	0	10	2	0
5	39	1	2	120	339	0	1	170	0	10	2	0
6	45	0	1	130	237	0	1	170	0	10	2	0
7	54	1	1	110	208	0	1	142	0	10	2	0
8	37	1	0	140	207	0	1	130	1	25	1	1
9	48	0	1	120	284	0	1	120	0	10	2	0

Figure 16: the data after encoding